Examen ARQ

Ejemplo1: ej. 3.2.3

Un determinado sistema permite el acceso virtual a 4 Gbytes de memoria y dispone de un mapa de memoria física de 1 Mbyte. El sistema virtual es paginado puro con tamaño de página de 4 kbytes. Para la traducción de una dirección virtual (DV) a una real (DR), dispone de una unidad TLB, común para código y datos, asociativa de 2 vías, que guarda un total de 32 direcciones virtuales. Se pide:

a) Los campos en los que se divide la DV para el acceso al TLB. Indicar la estructura y tamaño del TLB. Suponer que no existe ningún bit de control.

b) Suponiendo el TLB lleno, indicar razonando la respuesta, el número máximo de referencias a memoria que puede generar la CPU sin que se produzca un fallo de página en el TLB.

c) Suponiendo el TLB vacío, señalar el estado del mismo tras la ejecución del siguiente programa, suponiendo que el bucle se ejecuta cinco veces. En cada caso elegir cualquier dirección real válida.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Datos:

Memoria física: 1 Mbyte

Memoria virtual: 4 Gbytes

Tamaño Página: 4 Kbyte

TLB (común para código y datos, unificado), 2 vías, 32 direcciones virtuales

a)

4 Gbytes = 4 X 10243 = 232 -> DV 32 bits1 Mbyte = 1 X 10242 = 220 -> DR 20 bits

4 Kbytes = 4 X 1024 = 212 -> Offset 12 bits

DV -> 32 bits – 12 bits Offset = 20 bits número de página virtual

**DV = 20 bits número de página virtual + 12 bits Offset**

DR -> 20 bits – 12 bits Offset = 8 bits marco

**DR = 8 bits marco + 12 bits Offset**

32 direcciones virtuales y 2 vías = 16 entradas por vía = 24 = 4 bits índice

20 bits número de página virtual – 4 bits índice = 16 bits TAG

20 bits número de página virtual = 16 bits TAG + 4 bits índice

Campo TLB = TAG + descriptor = TAG + Marcos + Bits de control = 16 + 8 + 0 = 24

Descriptor = Marcos + Bits de control = 8 + 0 = 8 bits

Tamaño TLB = Directorio + Datos

Directorio = nº de vias X nº de entradas x TAG = 2 X 16 X 16 = 512 bits y 2 comparadores de 16 bits (nº de comparadores = vías, tamaño = TAG)

Datos = nº de vias X nº de entradas x descriptor = 2 X 16 X 8 = 256 bits

**Tamaño TLB = Directorio + Datos = 512 + 256 = 768 bits = 96 bytes**

b) Suponiendo TLB lleno

Número máximo de referencias a memoria = nº entradas X nº vias x direcciones página

Direcciones página = tamaño de página = 212

**Número máximo de referencias a memoria = 16 x 2 X** **212**

c)Suponiendo TLB vacio

DV: 00FFF000, se ha mapeado en la DR :32000

DV: 00000000, se ha mapeado en la DR :20000

DV: 00FF = TAG, F = índice, 000 = Offset

DR: 32 = marco, 000= Offeset

**Entrada: índice = F, TAG, 00FF, Marco = 32**

LOAD D1, $30H(A1) -> 00000030 -> 0000 = TAG, 0 = índice, 030 = Offset

**Entrada: índice = 0, TAG, 0000, Marco = 20**

Ejemplo2: ej. 3.2.13

Se dispone de un ordenador con un procesador cuyas direcciones virtuales son de 48 bits y que dispone de una memoria real de 4 Gbyte. Se opta por un sistema de memoria paginado, con un tamaño de página de 64 Kbytes y longitud del descriptor de página de 4 bytes en el que se incluye, el marco de página, un bit de presencia y varios bits de control más. Como la tabla de páginas es muy grande, ésta a su vez está paginada en tres niveles y sólo se guarda una parte en memoria real (los dos últimos niveles). La unidad de manejo de memoria (MMU) del sistema dispone de una unidad TLB 4-asociativo de 128 posiciones. Además del TLB, dispone de una memoria de sustitución directa, que contiene todos los descriptores del primer nivel de las páginas en las que se ha dividido la tabla de páginas. Se pide, justificando la respuesta, señalar:

a) Campos en que se divide la dirección virtual para el acceso al TLB y para el acceso a las tablas, la dirección real, los campos del TLB y los descriptores de página

b) El tamaño del TLB y de la memoria de sustitución directa presente en la MMU

c) El tamaño total de las tablas de páginas (solo nivel 2 y nivel 3). Si todas las páginas estuviesen en memoria que porcentaje representan respecto del total de memoria.

d) Cuantos accesos a memoria principal se deben realizar para obtener un dato, si la dirección no está en el TLB

Datos:

DV: direcciones virtuales son de 48 bits = 248 bytes

DR: 4 Gbyte = 232 bytes -> direcciones reales de 32 bits

Página: 64 Kbytes = 216 bytes -> 16 bits offset

Descriptor = 4 bytes = 32 bits

TLB 4-asociativo de 128 posiciones -> 128 posiciones en 4 vías

a)

DV: 48 bits; DR: 32 bits; Página: 64 Kbytes = 216 Bytes => 16 bits de offset En una tabla de páginas, del tamaño de una página, se pueden guardar 216/ 22 = 214 descriptores. Si hay 2 niveles de tabla en memoria queda para el primer nivel:

48 (DV) - 16(OFFSET) - 14(N3) - 14(N2) = 4 bits para el nivel 1 (N1).

Campos de la Dir.Virtual acc pag(48bits) -> Nivel 1: 4 bits + Nivel 2: 14 bits + Nivel 3: 14 bits + offset 16 bits = 48 bits

128/4 = 32 entradas por vía = 25 = 5 bits índice

48 bits DV – 16 bits offset – 5bits índice = 27 bits TAG

32 bits DR – 16 bits offset = 16 bits marco

Descriptor = 32 bits = Marcos + Bits de control = 16 + 16

Dirección virtual (48 bits, 32 bits número de página virtual) -> 27 bits TAG + 5 bits índice + 16 bits offset

Dirección Real (32 bits) -> 16 bits marco + 16 bits offset

Campos del TLB(59 bits) -> TAG + descriptor = TAG + Marcos + Bits de control = 27 + 16 + 16 = 59 bits

Campo de tabla pág. (32 bits) -> Marco 16 bits + 16 bits de control 32 bits

b)

Tamaño TLB = Directorio + Datos

Directorio = nº de vias X nº de entradas x TAG = 4 X 32 X 27 = 432 bytes y 4 comparadores de 32 bits (nº de comparadores = vías, tamaño = TAG)

Datos = nº de vias X nº de entradas x descriptor = 4 X 32 X 32 = 512 bytes

**Tamaño cache TLB = Directorio + Datos = 432 + 512 = 994 bytes**

La tabla de sustitución directa posee 24 (4vias) = 16 posiciones de 32 bits (128/4) = 512 bits = 64 bytes

c)

Tablas del N2: 24 x 214 x 22 = 16 tablas X 16.384 entradas X 4 Bytes = 220 Bytes

Tablas del N3: 218 x 214 x 22 = 262.144 tablas X 16.384 entradas X 4 Bytes = 234 Bytes Total = 234 + 220 = 16 GB + 1 MB = 16.385 MB

Representa aproximadamente el 234 / 232 \* 100 = 400% Es decir supera en un factor 4 la capacidad de la memoria real.

d)

Se necesita un mínimo de 3 accesos a memoria principal: La tabla de nivel 1 está presente en la MMU y no representa acceso a memoria. Luego un acceso a tabla de nivel 2, otro a la tabla del nivel 3 y finalmente el acceso al dato.

Ejemplo23: ej. 3.2.5

Un determinado sistema de memoria dispone de un TLB completamente asociativo con 16 entradas con tamaño de página de 256 bytes. El sistema dispone de una unidad caché de 1 Kbytes asociativa de 4 vías con dos palabras por bloque. El sistema soporta 262.144 (256k) direcciones virtuales. El bus de direcciones y el de datos son ambos de 16 bits. Se pide:

a) El tamaño del TLB y de la caché cuando la caché es real. Explicar cómo se obtiene el resultado.

b) El tamaño del TLB y de la caché cuando la caché es virtual. Explicar cómo se obtiene el resultado. NOTA: Considerar en ambos casos que no existen bits de control

Datos:

16 entradas

Tamaño de pagina 256 bytes

1Kbyte de cache

4 vías, 2 por bloque

DV = 256 Kbytes

Bus de direcciones y datos de 16 bits

Palabra = 2 bytes

DV = 256Kbytes = 218 -.> 18 bits de dirección virtual

DR= 216 = 16 bits dirección real

256bytes pag = 28 -> 8 bits offset

1Kbyte = 210 = 10 bits Numero de página bits marco

a)

**Dirección virtual = 8 bits offset + 10 bits TAG**

Tamaño TLB(asociativa) = 16 entradas por via X 18 bits = 288 bits

Dirección Real = 8 bits offset + 8 bits marco

Byte por bloque = 2 palabras por bloque = 22 bytes = 2 bits

Entradas = (1 Kbyte / bytes por bloque) / vias = (210 / 22)/ 4 = 28 / 4 = 26 entradas = 6 bits de índice

16 bits de dirección real– 6 bits índice – 2 bits byte por bloque = 8 bits TAG

**Direccion real(cache) = 8 bits TAG + 6 bits Indice + 2 bits B/B**

Directorio = nº de vias X nº de entradas x TAG = 4 X 64 X 8 = 256 bytes

Descriptor = 8 bits marco + 24 bits control

Datos = nº de vias X nº de entradas x descriptor = 4 X 64 X 32 = 1024 bytes

**Tamaño TLB = Directorio + Datos = 256 + 1024 = 1280 bytes**

Ejemplo 4

3.2.7. Un sistema ordenador de 32 bits, puede direccionar hasta 64Mbytes de memoria real, dispone además de las siguientes características: Espacio de direcciones virtuales de 1Gbyte con un tamaño de página de 4 Kbytes. TLB unificado completamente asociativo con 40 entradas y un byte de control por entrada. Unidad caché real y unificada de 20 Kbytes de capacidad, es asociativa por conjuntos con 5 vías, cada una con un tamaño de bloque de 32 bytes. Algoritmo de reemplazamiento LRU. Se pide:

a) Indicar en un esquema los campos de una entrada del TLB, indicando su tamaño. Calcular el tamaño total del TLB.

b) Indicar en un esquema los campos de una entrada de la unidad caché, indicando su tamaño. Calcular el tamaño total de la caché. No existen bits de control excepto los mínimos necesarios para el algoritmo LRU

c) Suponga una dirección virtual cualquiera que se encuentra en el TLB, explique cómo se calcula la correspondiente supuesta dirección real y con la ayuda de un esquema, indique como se accede a cada elemento de la caché, detallando que bits de la dirección se utilizan para extraer la palabra correcta.

Datos:

DR = 64Mbytes = 226 = 26 bits dirección real

DV = 1Gbyte = 240 = 20 bits dirección virtual

Tamaño de pagina = 4Kbytes = 212 = 12 bits offset

1 byte de control 0 8 bits

40 entradas

Cache real 20 Kbytes

5 vías

Tamaño bloque = 32 bytes

1. Estructura entrada a TLB

DV = 18 bits TAG + 12 bits Offset

DR = 14 bits de marco + 12 bits de offset

TLB = TAG + bits de control + marco = 18 +8 +14

Tamaño(asociativa) = 40 X (18 +8 +14) = 1600 bits = 400 bytes

1. Estructura a entrada a cache real:

20kbytes / 5 vias = 4Kbytes por via / 32 bytes/bloque = 128 bytes = 27 = 7 bits indice

Cache = 14 bits TAG (bits marco) + 7 bits Índice + 5 biots offset

Directorio = 128entradas X 5 vias x 14 TAG = 8960

Datos = 128entradas X ((5 vias x 256(bytes por bloque))) + 3 control) (descriptor) = 164224

Tmaño= 173184 bits = = 21.648 Bytes

1. Por ejemplo, tomemos 3B2CF0AA como dirección virtual, (sólo los 30 lsb). El offset es 0AA y la página virtual viene indicada por 3B2CF (sólo 18 lsb bits).

1.- Se compara 3B2CF con las 40 etiquetas que contiene el TLB y suponemos que se produce un acierto en una cierta entrada. Sea el marco de página al que se accede 2456 (14 lsb bits). Al concatenar con el offset se obtiene la dirección real, por ejemplo 2456 0AA (los 26 lsb bits). Con esta DR dividida en tres campos se accede a la caché: Etiqueta: 10100010011100, índice: 0000101 y byte en el bloque: 01010.

2.- En la caché, los bits de índice señalan la sexta entrada, en esa entrada y para las cinco vías se compara la etiqueta. Si hay acierto se selecciona el bloque correspondiente, y dentro de este bloque se envía el dato direccionado por los cinco últimos bits

Suponiendo que el acceso a MMU se da siempre antes de empezar el acceso a cachés, se pide calcular el tiempo medio de acceso a memoria (en ciclos) con los siguientes datos. Tiempo de acceso a TLB tTLB=1, tiempo de acceso a tablas ttablas=15, tiempo de acceso a caché L1 tL1=1, tiempo de acceso a caché L2 tL2=3, tiempo de acceso a memoria tmem=10. Los buses entre L1 y L2, y entre L2 y memoria son ambos de 4 Bytes de ancho, y el tiempo de acceso incluye acceder a todo el ancho de bus. La tasa de acierto local en L1 es del 90%, en L2 del 80% y en TLB del 98%, y se supone que nunca se falla en tablas de páginas. Si se falla en L1, traer un bloque a L1 desde L2 requiere un único acceso (4 Bytes). Por tanto, tB1=tL2=3. Pero si se falla en L2, traer un bloque a L2 desde memoria requiere 4 accesos (16 Bytes), por lo que esta penalización es de tB2=4·tmem=40 ciclos.

tacc = tMMU + tdato = (tTLB + fTLB·ttablas) + (tL1 + fL1·(tL2 + fL2·4·tmem)) = (1 + 0,02·15) + (1 + 0,1·(3 + 0,2·4·10)) = 1,3 + 2,1 = 3,4 ciclos.